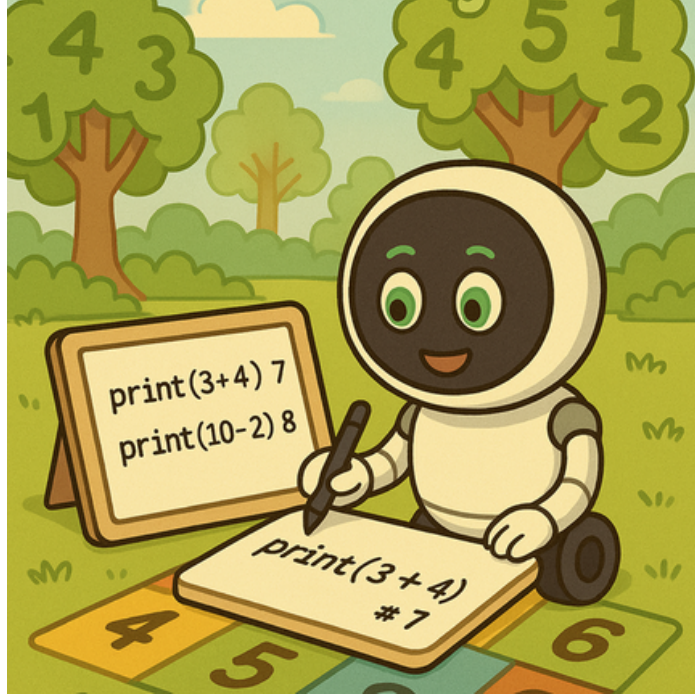


Numbers

Sahne 1

Orbit ile Sayılarla Tanışıyoruz!



Senaryo

Orbit artık yazı yazmayı öğrendi 📝

Şimdi sırada matematik var! 📊

Çünkü bir robot olmak için sadece konuşmak yetmez; hesap yapmayı da bilmek gerekir!

Bugün Orbit ile birlikte sayıları tanımayı ve onlarla işlem yapmayı öğreneceğiz! 🚀



Görev

- 🔴 İki sayı tanımla.
- 🔴 Bu sayılarla **toplama, çıkarma, çarpma ve bölme** işlemleri yap.
- 🔴 Sonuçları ekranda göster!

Numbers

```
# 1234 Sayılarla tanışıyoruz!
```

```
sayi1 = 10
```

```
sayi2 = 5
```

```
# + Toplama
```

```
toplam = sayi1 + sayi2
```

```
print("🇧🇷 Toplam:", toplam)
```

```
# - Çıkarma
```

```
fark = sayi1 - sayi2
```

```
print("🇧🇷 Fark:", fark)
```

```
# × Çarpma
```

```
carpim = sayi1 * sayi2
```

```
print("🇧🇷 Çarpım:", carpim)
```

```
# ÷ Bölme
```

```
bolum = sayi1 / sayi2
```

```
print("🇧🇷 Bölüm:", bolum)
```

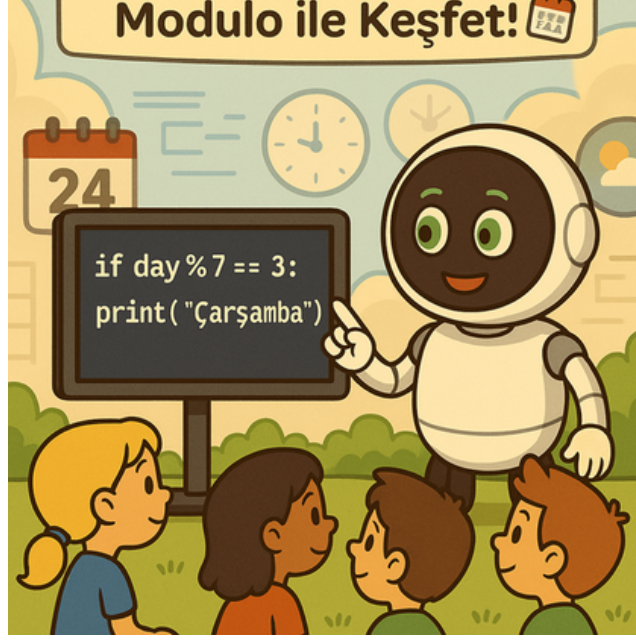


Kazanım

- ✓ Sayıları tanımlamayı öğrendim ¹²₃₄
- ✓ Matematiksel işlemleri Python'da yapabildim $+-\times\div$
- ✓ Robotum Orbit artık hesap yapabiliyor! 🤖🧠
- ✓ `print()` ile sayıları ekrana yazdırmayı pekiştirdim 💻

Numbers

Sahne 2

Orbit ile Modulo Operatörüyle
Günleri Hesaplıyoruz! 📅🤖

Senaryo

Orbit, haftanın günlerini hesaplamak istiyor! 🤔

Mesela, ayın 3. günü Çarşamba ise, 7 gün sonra yani 10. gün, 17. gün ve 24. gün de **Çarşamba olur!** 🎉

Çünkü haftada 7 gün vardır ve 7 gün sonra aynı gün tekrar başlar! 🔄

İşte burada **modulo (%) operatörü** devreye giriyor! ⚙️🧠



Görev

📌 Ayın 3., 10., 17. ve 24. günlerinin haftanın hangi günü olduğunu bul. 🔍

📌 % 7 işlemini kullan! 📅

📌 Sonuçları ekranda göster! 💻

Numbers

```
# Ayın ilk günü Çarşamba (gün sayısı )
```

```
ilk_gun = 3
```

```
print("📅 3 % 7 =", 3 % 7)
```

```
print("📅 10 % 7 =", 10 % 7)
```

```
print("📅 17 % 7 =", 17 % 7)
```

```
print("📅 24 % 7 =", 24 % 7)
```

```
print("\n🎉 Gördün mü? Mod 7 işleminin sonucu hep 3 çıktı! 🚀")
```

```
print("🎯 Bu da demek oluyor ki 3., 10., 17. ve 24. günler hep aynı gün,  
yani Çarşamba! 📅 17 ✅")
```



Kazanım

- ✅ % operatörü ile kalan sayıyı bulmayı öğrendim! 🎉
- ✅ Haftanın günlerini kolayca hesaplayabiliyorum! 📅
- ✅ Robotum Orbit artık tarihleri süper hızlı hesaplıyor! 🤖⚡
- ✅ Python'da basit matematik işlemleri yapabiliyorum! 💻✨

Numbers

Sahne 3

Orbit ile Sayılar ve Metinler (String)
Arasındaki Farkı Öğreniyoruz!

Senaryo

Orbit, ekrana yazı yazmayı ve sayıları yazdırmayı öğrendi!

Ama dikkat! "20" ve 20 aynı şey değil!

Birisi **metin (string)**, diğeri ise **sayı (integer)**.

Bir de "20" + "20" ile 20 + 20 farklı şey yapar! Gel birlikte bakalım! 🚀



Görev

- 🔴 print("20") ne gösterir?
- 🔴 print(20 + 20)ne gösterir?
- 🔴 print("20" + "20") ne gösterir?
- 🔴 Sonuçları açıklayalım!

Numbers

```
print("20")          # Metin olarak 20 yazdırır: 20
print(20 + 20)        # Sayılar toplandığı için sonuç 40 olur
print("20" + "20")    # Metinleri yan yana ekler, sonuç: 2020
print(20 * 2)         # Sayıları çarptığı için sonuç 40 olur.
print("20" + "*" + "2") # Metinleri yan yana ekler, sonuç: 20 * 2
```



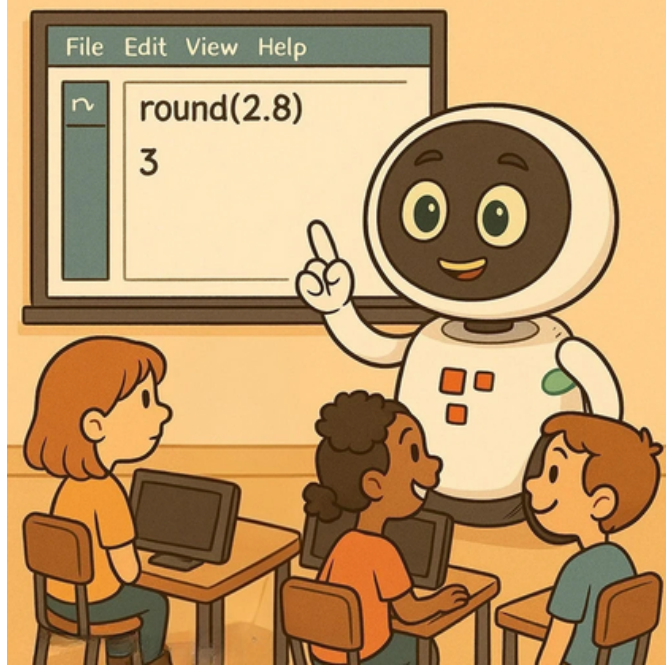
Kazanım

- ✓ Metin ile sayının farkını öğrendim! abc vs $\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}$
- ✓ + operatörünün sayılarla toplama, metinlerle birleştirme yaptığını anladım! + vs
- ✓ Robotum Orbit artık ekranı daha iyi kullanıyor!

Numbers

Sahne 4

Orbit ile Sayıları Yuvarlama 12 34 56 78 90



Senaryo

Orbit matematikte biraz daha ileri gidiyor!

Bazen sayıları yuvarlamak gerekiyor, mesela 3.14'ü 3 yapabiliriz.

Hadi Orbit'le bunu deneyelim! 🚀



Görev1: Sayıyı yuvarla!

- ✦ Bir ondalıklı sayı tanımla.
- ✦ **round()** fonksiyonuyla en yakın tam sayıya yuvarla.
- ✦ Sonucu ekrana yazdır! 📄

Numbers

```
sayi1 = 3.14159  
print("Orijinal sayı:", sayi1)  
  
# Sayıyı yuvarla  
yuvarlanmis = round(sayi1)  
print("Yuvarlanmış sayı:", yuvarlanmis)
```



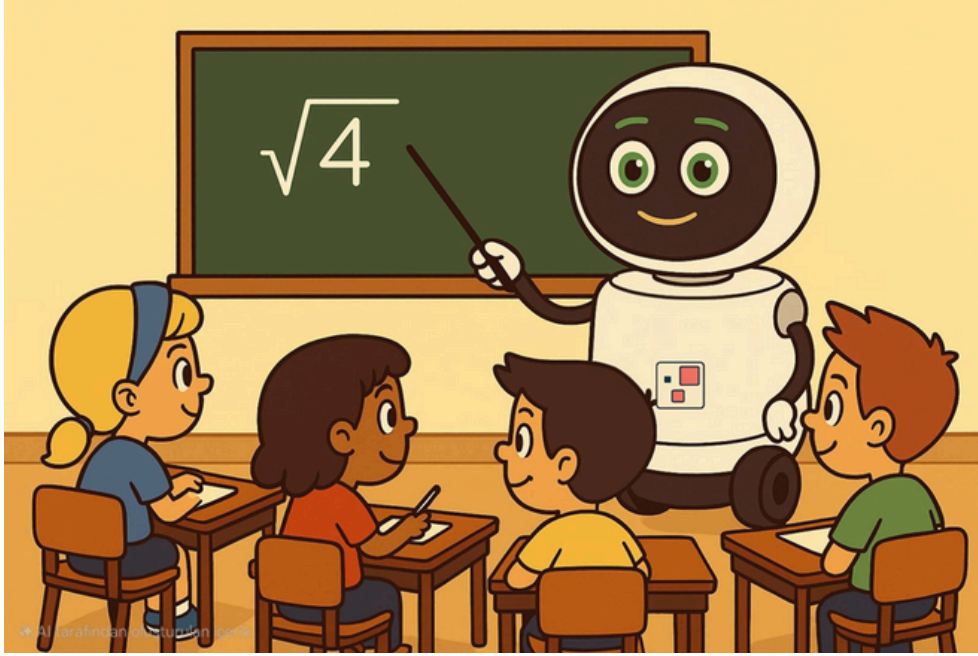
Kazanım

- ✓ Ondalıklı sayıları nasıl yuvarlayacağımı öğrendim!
- ✓ Python'da round() fonksiyonunu öğrendim!

Numbers

Sahne 5

Orbit ile Karekk Buluyoruz! 🇧🇪 🤖 🌟



Senaryo

Bugn Orbit ile **karekk** bulmayı ğreneceğiz!

Karekk ne demek? Mesela 4'n karekk 2'dir nk $2 \times 2 = 4$! 🤔

Bir sayının karekk, kendisiyle arpıldığında o sayıyı veren sayıdır!

Python'da bu iřlem ok kolay! Hadi deneyelim! 🚀 📐




Grev

- ✦ Bir sayı tanımla. `1234`
- ✦ `math.sqrt()` fonksiyonunu kullanarak karekkn hesapla.
- ✦ Sonucu ekrana yazdır. 🖨
- ✦ Orbit'e bu sayının hangi sayıdan oluřtuğunu gster! 🤖

Numbers



 **Not:** Eğer karekökünü almak istediğin sayı tam kare değilse, sonuç ondalıklı olabilir!

Örneğin 20'nin karekökü yaklaşık olarak 4.47 olur! 🔍

```
sayi1 = 3.14159
```

```
import math # 📦 Matematik kütüphanesini ekliyoruz
```

```
sayi = 16 # 📄 12/34 Kareköklü sayı
```

```
karekok = math.sqrt(sayi) # 🧠 Karekökünü hesapla
```

```
print("📄 12/34 Sayının kendisi:", sayi)
```

```
print("📄 Sayının karekökü:", karekok)
```



Kazanım

- ✓ Bir sayının karekökünü bulmayı öğrendim! 📄
- ✓ math.sqrt() fonksiyonunu kullandım! 📄
- ✓ Python'da farklı matematik işlemlerini yapabiliyorum! 💡
- ✓ Orbit artık bilim robotu gibi hesap yapıyor! 🤖 📊